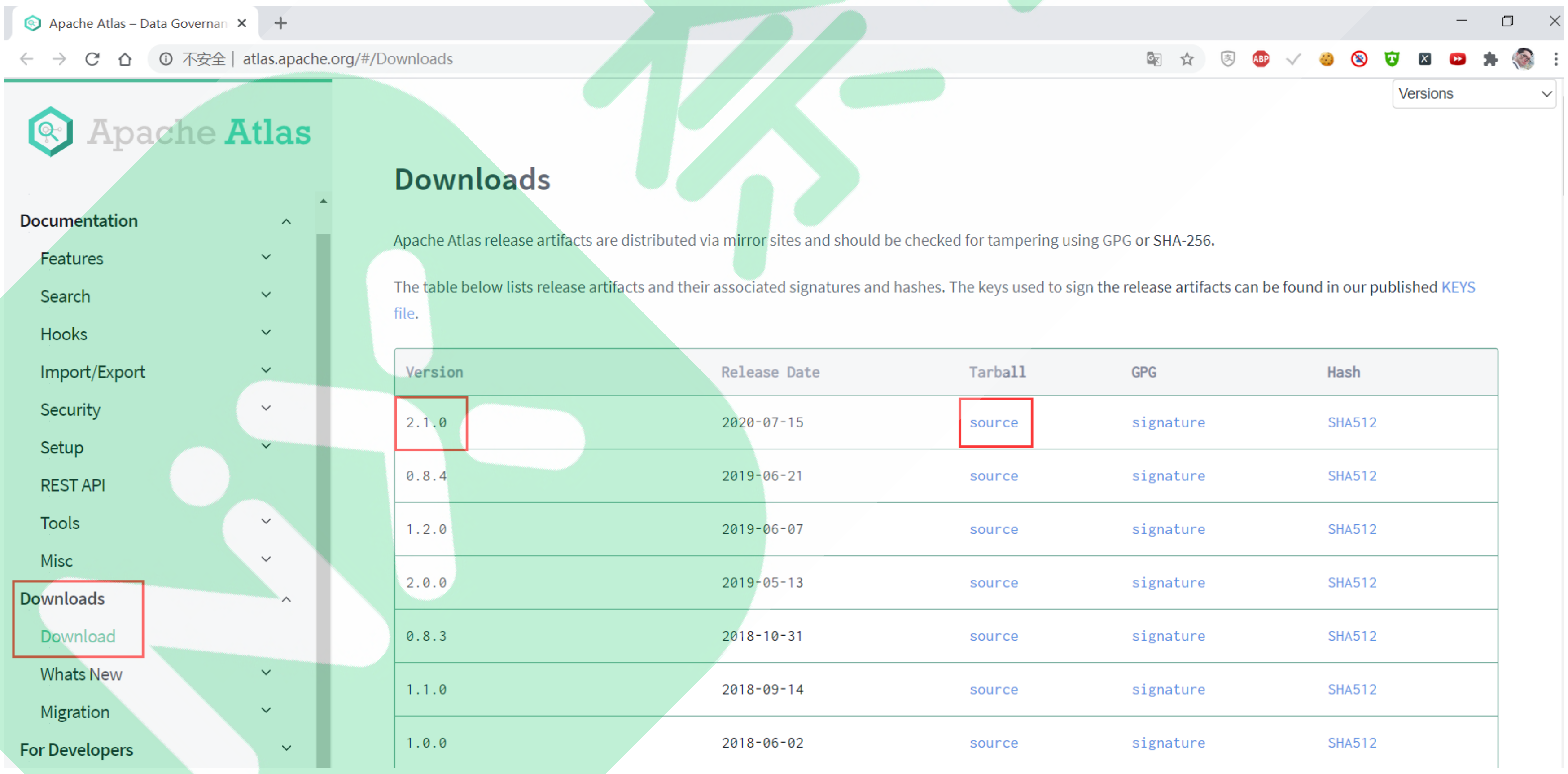
1. 安装 Atlas

1.1. 下载 atlas

官网：http://atlas.apache.org/#/Downloads



Downloads

Apache Atlas release artifacts are distributed via mirror sites and should be checked for tampering using GPG or SHA-256.

The table below lists release artifacts and their associated signatures and hashes. The keys used to sign the release artifacts can be found in our published KEYS file.

Version	Release Date	Tarball	GPG	Hash
2.1.0	2020-07-15	source	signature	SHA512
0.8.4	2019-06-21	source	signature	SHA512
1.2.0	2019-06-07	source	signature	SHA512
2.0.0	2019-05-13	source	signature	SHA512
0.8.3	2018-10-31	source	signature	SHA512
1.1.0	2018-09-14	source	signature	SHA512
1.0.0	2018-06-02	source	signature	SHA512

1.2. 编译

见编译文档

如果不自行编译的话， 也可以直接下载安装包！

1.3. 准备操作

1.3.1. 安装 zookeeper

如果是内嵌版，则不需要安装

1.3.2. 安装 kafka

如果是内嵌版，则不需要安装

1.3.3. 安装 hbase

如果是内嵌版，则不需要安装

1.3.4. 安装 solr

如果是内嵌版，则不需要安装

1.4. 上传到服务器

```
put -r "apache-atlas-2.1.0-bin.tar.gz"
```

1.5. 解压缩安装

```
tar -zxvf ~/soft/apache-atlas-2.1.0-bin.tar.gz -C ~/apps/
```

1.6. 配置环境变量

```
vim ~/.bashrc
```

```
export ATLAS_HOME=/home/bigdata/apps/apache-atlas-2.1.0
export PATH=$PATH:$ATLAS_HOME/bin
```

```
source ~/.bashrc
```

1.7. 修改配置文件

修改：atlas-env.sh 中的 java 和 hbase 等相关配置

```
cd $ATLAS_HOME
```

```
vim conf/atlas-env.sh
```

```
export JAVA_HOME=/usr/local/java/jdk1.8.0_181
export MANAGE_LOCAL_HBASE=true       (如果要使用外部的zk和hbase，则改为false)
export MANAGE_LOCAL_SOLR=true        (如果要是用外部的solr，则改为false)
```

增加一项配置，把 hbase-site.xml 配置文件包含进来：

```
export HBASE_CONF_DIR=$HBASE_HOME/conf
```

修改 atlas 具体的存储组件相关配置：

```
cd $ATLAS_HOME
vim conf/atlas-application.properties
```

```
# Hbase地址（对应的zk地址）配置（自带hbase会根据此端口启动一个zk实例）
# 如果使用外部hbase，则填写外部zookeeper地址
atlas.graph.storage.hostname=bigdata02:2181,bigdata03:2181,bigdata04:2181/kafka251
atlas.graph.storage.backend=hbase
atlas.graph.storage.hbase.table=apache_atlas_janus
export HBASE_CONF_DIR=/opt/atlas/conf/hbase/conf

atlas.EntityAuditRepository.impl=org.apache.atlas.repository.audit.HBaseBasedAuditRepository

# Solr地址配置
atlas.graph.index.search.solr.http-urls=http://localhost:8983/solr
```

```
# Kafka相关配置
atlas.notification.embedded=true                    # 如果要使用外部的kafka，则改为false
# 内嵌kafka会根据此端口启动一个zk实例
atlas.kafka.zookeeper.connect=bigdata02:2181,bigdata03:2181,bigdata04:2181/kafka251  # 如果使用外部kafka，则填写外部zookeeper地址
atlas.kafka.bootstrap.servers=bigdata02:9092,bigdata03:9092,bigdata04:9092           # 如果使用外部kafka，则填写外部broker server地址
```

1.8. 拷贝 core-site.xml 和 hdfs-site.xml

如果是 HA 的 Hadoop 集群，拷贝 Hadoop 集群的配置文件，到 Atlas 中：

```
cd $HADOOP_HOME/etc/hadoop
cp core-site.xml hdfs-site.xml $ATLAS_HOME/conf
```

1.9. 拷贝 hbase-site.xml

拷贝 HBase 的配置文件到 Atlas：

```
cd ~/apps/apache-atlas-2.1.0
cp $HBASE_HOME/conf/hbase-site.xml $ATLAS_HOME/conf
```

```
mkdir -p $ATLAS_HOME/hbase/conf
cp $HBASE_HOME/conf/hbase-site.xml $ATLAS_HOME/hbase/conf/
```

1.10. 手动启动 hbase

如果要使用外置 HBase，则在启动 Atlas 之前，必须先启动 HBase：

```
cd $ATLAS_HOME/hbase
```

```
$ATLAS_HOME/hbase/bin/start-hbase.sh
$ATLAS_HOME/hbase/bin/hbase-daemon.sh start regionserver
```

访问 webui：http://bigdata05:16010 或者 http://bigdata02:16010

1.11. 手动启动 solr

如果要使用外置 solr，则在启动 Atlas 之前，必须先启动 solr：

```
# 进入atlas自带的solr目录
cd $ATLAS_HOME/solr

# 启动solr
bin/solr start -c -z localhost:2181 -p 8984 -force                  #启动solr
```

第一次安装 Atlas 并启动 Solr 的话，记得，要手动创建这个索引：

```
# 创建初始化collections
cd ~/apps/solr-7.7.3/
bin/solr create -c vertex_index -shards 1 -replicationFactor 1 -force
bin/solr create -c edge_index -shards 1 -replicationFactor 1 -force
bin/solr create -c fulltext_index -shards 1 -replicationFactor 1 -force
```

1.12. 手动启动 kafka

启动 Kafka，并手动创建两个 kafka topic：

```
kafka-topics.sh --zookeeper bigdata02:2181,bigdata03:2181,bigdata04:2181/kafka251 --create --replication-factor 3 --partitions 3 --topic ATLAS_HOOK
kafka-topics.sh --zookeeper bigdata02:2181,bigdata03:2181,bigdata04:2181/kafka251 --create --replication-factor 3 --partitions 3 --topic ATLAS_ENTITIES
```

1.13. 启动 Atlas

启动命令：

```
cd $ATLAS_HOME
bin/atlas_start.py
```

停止命令：

```
~/apps/apache-atlas-2.1.0/bin/atlas_stop.py
```

由于第一次启动事件比较长，大家如果有兴趣的话，可以观看启动日志，来了解 Atlas 第一次启动会做些什么操作：

```
cd /home/bigdata/apps/apache-atlas-2.1.0/logs
tail -f application.log
```

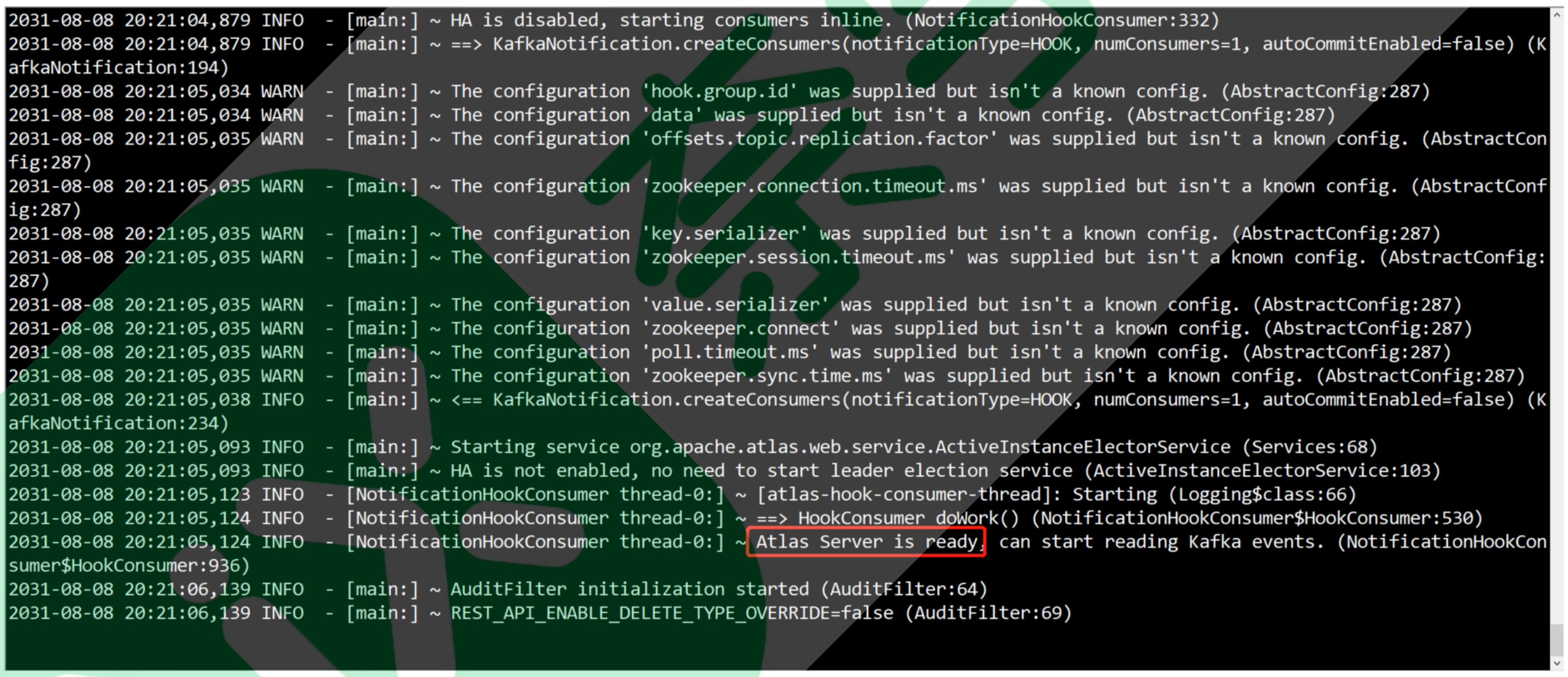
1.14. 启动成功

启动成功提示：~/apps/apache-atlas-2.1.0/bin/atlas_start.py

```
[bigdata@bigdata05 apache-atlas-2.1.0]$ ~/apps/apache-atlas-2.1.0/bin/atlas_start.py
starting atlas on host localhost
starting atlas on port 21000
...............................................................................................
Apache Atlas Server started!!!
```

第一次启动，需要花费很长很长的时间，具体时间跟 HBase, Hive 等的表的数量来决定。请耐心等待。可以关注日志的情况。

```
cd /home/bigdata/apps/apache-atlas-2.1.0/logs
tail -f application.log
```



```
2031-08-08 20:21:04,879 INFO  - [main:] ~ HA is disabled, starting consumers inline. (NotificationHookConsumer:332)
2031-08-08 20:21:04,879 INFO  - [main:] ~ ==> KafkaNotification.createConsumers(notificationType=HOOK, numConsumers=1, autoCommitEnabled=false) (K
afkaNotification:194)
2031-08-08 20:21:05,034 WARN  - [main:] ~ The configuration 'hook.group.id' was supplied but isn't a known config. (AbstractConfig:287)
2031-08-08 20:21:05,034 WARN  - [main:] ~ The configuration 'data' was supplied but isn't a known config. (AbstractConfig:287)
2031-08-08 20:21:05,035 WARN  - [main:] ~ The configuration 'offsets.topic.replication.factor' was supplied but isn't a known config. (AbstractCon
fig:287)
2031-08-08 20:21:05,035 WARN  - [main:] ~ The configuration 'zookeeper.connection.timeout.ms' was supplied but isn't a known config. (AbstractConf
ig:287)
2031-08-08 20:21:05,035 WARN  - [main:] ~ The configuration 'key.serializer' was supplied but isn't a known config. (AbstractConfig:287)
2031-08-08 20:21:05,035 WARN  - [main:] ~ The configuration 'zookeeper.session.timeout.ms' was supplied but isn't a known config. (AbstractConfig:
287)
2031-08-08 20:21:05,035 WARN  - [main:] ~ The configuration 'value.serializer' was supplied but isn't a known config. (AbstractConfig:287)
2031-08-08 20:21:05,035 WARN  - [main:] ~ The configuration 'zookeeper.connect' was supplied but isn't a known config. (AbstractConfig:287)
2031-08-08 20:21:05,035 WARN  - [main:] ~ The configuration 'poll.timeout.ms' was supplied but isn't a known config. (AbstractConfig:287)
2031-08-08 20:21:05,035 WARN  - [main:] ~ The configuration 'zookeeper.sync.time.ms' was supplied but isn't a known config. (AbstractConfig:287)
2031-08-08 20:21:05,038 INFO  - [main:] ~ <== KafkaNotification.createConsumers(notificationType=HOOK, numConsumers=1, autoCommitEnabled=false) (K
afkaNotification:234)
2031-08-08 20:21:05,093 INFO  - [main:] ~ Starting service org.apache.atlas.web.service.ActiveInstanceElectorService (Services:68)
2031-08-08 20:21:05,093 INFO  - [main:] ~ HA is not enabled, no need to start leader election service (ActiveInstanceElectorService:103)
2031-08-08 20:21:05,123 INFO  - [NotificationHookConsumer thread-0:] ~ [atlas-hook-consumer-thread]: Starting (Logging$class:66)
2031-08-08 20:21:05,124 INFO  - [NotificationHookConsumer thread-0:] ~ ==> HookConsumer doWork() (NotificationHookConsumer$HookConsumer:530)
2031-08-08 20:21:05,124 INFO  - [NotificationHookConsumer thread-0:] ~ Atlas Server is ready, can start reading Kafka events. (NotificationHookCon
sumer$HookConsumer:936)
2031-08-08 20:21:06,139 INFO  - [main:] ~ AuditFilter initialization started (AuditFilter:64)
2031-08-08 20:21:06,139 INFO  - [main:] ~ REST_API_ENABLE_DELETE_TYPE_OVERRIDE=false (AuditFilter:69)
```

1.15. 检测是否正常

检查进程

```
[bigdata@bigdata05 apache-atlas-2.1.0]$ jps
3008 jar
3824 Jps
3730 Atlas
3347 HMaster
3529 HRegionServer
2764 DataNode
2669 QuorumPeerMain
```

检测端口占用：

```
[bigdata@bigdata05 apache-atlas-2.1.0]$ netstat -nltp
```

结果如下：

```
tcp        0      0 0.0.0.0:21000           0.0.0.0:*               LISTEN      3730/java
```

1.16. 访问 web ui

启动成功后，浏览器输入：http://bigdata05:21000 ，默认用户名密码为 admin/admin

1.17. Rest API 测试

启动后，测试：

```
curl -u admin:admin http://bigdata05:21000/api/atlas/admin/version
```

1.18. 运行示例数据

```
~/apps/apache-atlas-2.1.0/bin/quick_start.py
```

输入用户名和密码：admin admin